Achieving k-anonymity Using Parallelism in Full Domain Generalization

Kodam Sai Kumar

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India
May 2015

Achieving k-anonymity Using Parallelism in Full Domain Generalization

Thesis submitted in partial fulfillment of the requirements for the degree of

Master of Technology

in

Computer Science and Engineering

(Specialization: Information Security)

by

Kodam Sai Kumar

(Roll- 213CS2157)

Under the supervision of

Prof. Korra Sathya Babu

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela, Odisha, 769 008, India

May 2015

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Certificate

This is to certify that the work in the thesis entitled ***Achieving k-anonymity Using Parallelism in Full Domain Generalization*** by ***Kodam Sai Kumar*** is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Master of Technology with the specialization of Information Security in the department of Computer Science and Engineering, National Institute of Technology Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Place: NIT Rourkela
Date: May 28, 2015

Prof. Korra Sathya Babu
Professor, CSE Department
NIT Rourkela, Odisha

Acknowledgment

I am grateful to numerous local and global peers who have contributed towards shaping this thesis. At the outset, I would like to express my sincere thanks to Prof. Korra Sathya Babu for his advice during my thesis work. As my supervisor, he has constantly encouraged me to remain focused on achieving my goal. His observations and comments helped me to establish the overall direction to the research and to move forward with investigation in depth. He has helped me greatly and been a source of knowledge.

I am very much indebted to Prof. Santanu Ku. Rath, Head-CSE, for his continuous encouragement and support. He is always ready to help with a smile. I am also thankful to all the professors at the department for their support.

I would like to thank Mr. Pramit Mazundar for his encouragement and support. His help can never be penned with words.

I would like to thank all my friends and lab-mates for their encouragement and understanding. Their help can never be penned with words.

I must acknowledge the academic resources that I have got from NIT Rourkela. I would like to thank administrative and technical staff members of the Department who have been kind enough to advise and help in their respective roles.

Last, but not the least, I would like to dedicate this thesis to my father and mother, for their love, patience, and understanding.

Kodam Sai Kumar

Roll-213cs2157

Declaration

I Kodam Sai Kumar of Computer Science and Engineering with Roll no 213CS2157 hereby declare that the project submitted by me is solely of my work and is not copied from any other source where ever may available and It has not been previously submitted for any academic degree. I had verified my thesis report through Turnitin software for plagiarism. All sources of quoted information have been acknowledged by means of appropriate references.

If in future my work was found to be plagiarized from any other persons work, then in that situation I alone will be responsible for it.

Date: May 28, 2015

Kodam Sai Kumar
NIT Rourkela

Abstract

Preserving privacy while publishing data has emerged as key research area in data security and has become a primary issue in publishing person specific sensitive information. How to preserve one's privacy efficiently is a critical issue while publishing data. k- Anonymity is a key technique for de-identifying the sensitive datasets. In our work, we have described an approach to implement various k-anonymity algorithms and also propose a parallelism method that produces better results with the real-world datasets. Additionally, we suggest a new approach that attains better results by applying a parallelism approach and exploiting various characteristics of our suggested approach. The proposed approach uses the concept of samarati algorithm to generalize the lattice and uses the binary search method. The proposed algorithm generates the levels using binary search in the lattice and then uses the parallel mechanism for evaluating the nodes. The proposed algorithm has less execution time than other full domain generalization algorithms for k-anonymization.

Key words: k-Anonymity, Parallelism, Full Domain Generalization, Quasi-Identifier.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over last 20 years, the digitization of our daily lives has led to an increase in the data collected by individuals, corporations, and governments. This digitally available data (known as microdata) has created a good opportunity for decision making based on available information. Because of mutual benefits, or by organization's policies, publication of digitally available data is required to improve decision making. But the collected microdata in its native form may contain person specific sensitive information of individuals whose privacy can be violated if the original data is published.

So the important task is to protect the privacy of this microdata. There exists some guidelines, agreements and policies about how and what data should be published so that the data remains useful for research and analysis and at the same, individual's privacy is preserved, referred as privacy preserving data publishing (PPDP) [1].

Privacy preserving data publishing (PPDP) is an approach to publish practically useful data without violating individuals privacy. PPDP focuses on data anonymization that attempt to conceal the identity of record holders, considering that private data must be maintained for data analysis [2]. PPDP consist of two phases: Data collection and data publication.

1. Data collection: in this phase, the original data from record holders is retrieved by the data publisher.

2. Data publishing: in this phase, the data retrieved by record holders in data

collection phase, is released to data recipient for analysis and mining purpose. A real time scenario of PPDP is given as follows:

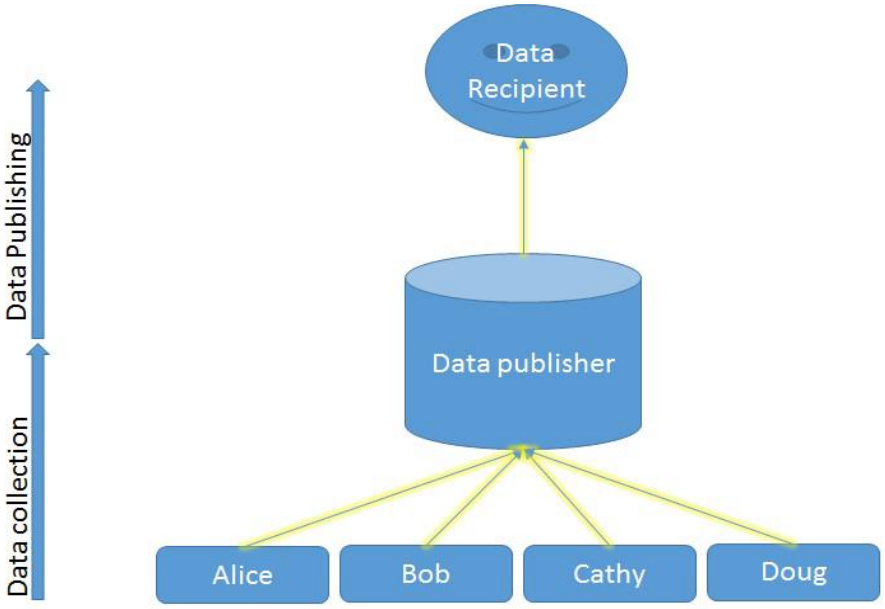

Figure 1.1: Privacy Preserving Data Publishing

In this example, we can compare this with the hospital patient scenario where Alice, Bob, Cathy, Doug are the patient (Record holders) and data publisher (hospital) collects the information from record owners and gives it medical center (Data recipient) for research and analysis purpose [3]. Finally data recipient perform data mining to retrieve useful information. On the basis of trust level the data publisher is categorizes in two models: trusted data publisher and untrusted data publisher.

1. Trusted Data publisher: In this model, the record holders know that the data publisher is reliable and they are providing their personal information for the analysis [4].

2. Untrusted data publisher: In it, the publisher may not be reliable and may try to gain confidential information from the record holders.

In this thesis, we consider only preserving of information privacy, which protects sensitive information from being brought to the attention of others. Privacy preserving is the ability to limit the diffusion and use of one's personal data. Privacy can refer to an individual where nobody should know about any entity after performing data mining or an organization to protect knowledge about a collection of entities. Various approaches followed for individual privacy preserving are data obfuscation, value swapping, perturbation, etc. Each organization adopts a framework for disclosing individual entity values to the public.

1.1 Motivation

Nowadays, clinical institutions are increasingly asked to make their raw, non-aggregated data (also called microdata), electronically available for research purposes. However, since such data may contain private personal information as in the case of medical records, the identity of the entities involved must remain confidential.

A telephone poll has been conducted in the U.S. in which 88% of the respondents replied that to the best of their knowledge, no medical data about themselves had ever been given without their permission. In a second question, 87% said laws should prohibit organizations from providing out medical data without obtaining the patients permission. Thus, the public would prefer that only employees and directly involved persons have authority to their records and that these people be bound by the strict ethical and legal standards that prohibit's further disclosure [5].

Nowadays, the disclosure of health data is strictly regulated in many jurisdictions, and institutions are often legally required to apply privacy-enhancing transformations to health data prior to their disclosure to researchers. For example, the Health Insurance Portability and Accountability Act (HIPAA) [6] in the U.S., and the Personal Health Information Protection Act (PHIPA) [7] in Canada, are the some of well-recognized privacy regulations, protects the confidentiality of electronic healthcare data.

In order to provide the privacy of the respondents to which the data refer, released data were at first de-identified by removing all the explicit identifiers such as phone numbers, addresses and names. However this de-identified data could still have other implicit identifying characteristics such as sex, birth date, race and postal code which, when considered all together, can uniquely, or almost uniquely pertain to the specific individuals. These sets of characteristics are often called *quasi-identifiers*.

For instance, in one study, Sweeney estimated that 87.1% of the United States population can be uniquely recognized by the combination of the date of birth, gender, 5-digit ZIP code because such information can be linked to public od free available databases such as driving records and voter list. To prove her point, Sweeney re-identified a series of supposedly anonymous medical data including one data, which belonging to William Weld the governor of Massachusetts at the time using a voter list she bought from the city of Cambridge, Massachusetts for a mere $20 [8] [5].

To illustrate the concept, consider 1.1, which exemplifies medical data to be released. In this table, data have been de-identified by suppressing names and Social Security Numbers (SSNs) so not to explicitly disclose the identities of patients.

Table 1.1: Deidentified private table (medical data)

SSN.	**Name**	**Race**	**DOB**	**SEX**	**ZIP**	**Marital Status**	**Disease**
		asian	64/04/12	F	94142	divorced	hypertension
		asian	64/09/13	F	94141	divorced	obesity
		asian	64/04/15	M	94140	married	chest pain
		asian	63/06/13	F	94139	married	obesity
		asian	63/06/18	M	94139	married	short breath
		black	65/08/27	F	94138	single	short breath
		black	64/08/27	F	94139	single	obesity
		white	65/08/27	M	94139	single	chest pain
		white	64/08/27	M	94141	widow	short breath

However, notice that there is only one divorced female (F) born on 64/04/12 and living in the 94142 area. This combination, if unique in publicly available databases such as in Table- 1.2, identifies the corresponding tuple as pertaining to

Table 1.2: Non-de-identified publicly available table

Name.	**Address**	**City**	**ZIP**	**DOB**	**Sex**	**Marital Status**
–	–	–	–	–	–	–
–	–	–	–	–	–	–
Alex	120 PK Street	Texas	94142	64/04/12	F	divorced
–	–	–	–	–	–	–
–	–	–	–	–	–	–

Alex, 120 PK Street, Texas, thus revealing that she has reported hypertension [9].

In order to overcome the potential for a privacy breach, some researchers tried to further de-identify the data by using techniques such as scrambling and swapping values and adding noise to the data while maintaining an overall statistical property of the result. However, this compromised the integrity, or truthfulness, of the information released [1].

In a different direction, intensive research has been directed towards the anonymization of the data. Although guaranteeing complete anonymity is obviously an impossible task, the k-anonymity concept has been introduced: "A data release is said to satisfy k-anonymity if every combination of values of quasi-identifiers can be distinctly matched to at least k individuals in that release".

1.2 Objective

Several algorithms were developed with the purpose of making de-identified data k-anonymous [10], hence readily available for researchers. However, we are only concerned with the methods that aim to achieve k-anonymity through full domain global recoding, hierarchical generalization and minimal suppression, as will be motivated in the next chapter. Mainly, two of the most popular approaches that fall under the former specifications and that were heavily used so far for clinical data are Sweenys Datafly algorithm and Samaratis algorithm.

So far no one has empirically evaluated these algorithms in order to recognize which does a better job in balancing satisfactory privacy with minimum informa-

tion loss, or how their solution compares with respect to the optimal one. More importantly, these approaches always rely on some heuristic that would approximate a "good" solution rather than actually finding the optimal one with respect to any given preference or information loss metrics.

Other existing methods, such as Incognito, tend to find all the possible solutions. However, a major drawback of such approaches is that the number of solutions they return is usually very high, and it is impractical to check the information loss of all of them in order to find the optimal one.

Resolving the above issues is very important. Accordingly, by assuring better solutions, researchers will benefit immensely, since the better the quality of the anonymized data, and the less the information loss, the more valuable that data is for their research. Therefore, our objective is to evaluate these algorithms and determine whether a better one can be devised in order to efficiently find an optimal solution.

1.3 Thesis Contribution

The contribution of this thesis is two field: criteria:

1. First, we implemented the following algorithms
    - Samarati Algorithm
    - OLA Algorithm and
    - Incognito Algorithm

2. Second, we propose our own approach, a new method to find efficiently an optimal solution.

1.4 Thesis Organization

- **Chapter-1**, In this chapter we explore briefly about data publishing and what is privacy preserving, why there is need of privacy preserving techniques while publishing data. How anonymization can be used to preserve privacy .To maintain privacy a model k-anonymity is explained in it and its basic details and attack on this model.

- **Chapter-2** In this chapter we have discussed, metric that are used to calculate the quality of anonymized data, the previous algorithms that have been used for k-anonymization.

- **Chapter-3**, In this chapter we explained that to achieve k-anonymity, the best way is to find the lattice in the parallelism manner using minimum information loss to obtain the local optimal node.

- **Chapter-4**, In the chapter we have plotted the graph, for different values of k taken execution time vs quasi-identifier and distortion vs quasi-identifier. We compare and analysis the results of our approach with previous algorithms.

- **Chapter-5**, In this chapter, we have explained that after comparing the results and analysis we can conclude that our purposed algorithm gives takes less time than other efficient algorithms while other metric also gives better results in maximum cases.

Chapter 2

Literature Survey

2.1 Preliminary Concepts and Definitions

In what follows, we assume the existence of an already de-identified private1 table PT to be anonymized. The rows in PT may be referred to as tuples, and the table is assumed to have at least k tuples. Moreover, the columns in the table are the attributes, and unless otherwise mentioned, the set of PT's attributes will be strictly considered as the quasi-identifier.

In basic scenario of privacy preserving data publishing, the published data table has the following form:

DT (Explicit Identifier, Quasi-Identifier, Sensitive Attributes, Non-Sensitive Attributes)

Table 2.1: Identifier's

Identifier	**Quasi-Identifier**			**Sensitive**
Name	**Birth Date**	**Sex**	**Zipcode**	**Disease**
Alice	21/01/79	Male	52368	Flu
Beth	15/11/81	Female	56478	Hepatitis
Carol	23/05/79	Female	52314	AIDS
Dan	06/12/84	Male	50301	Canser
Ellen	20/09/83	Female	57612	Fever

2.1.1 Explicit Identifier

It is a group of attributes (for e.g. voter id, Name etc.), able to identify individual record explicitly.

2.1.2 Quasi-Identifier

A group of attributes from a table whose combination can be used to identify some other record from dataset. Quasi-identifiers may be used to re-identify an individual record from the table. For example [2] combination of (Job, Postcode, and Date of birth) of all these attribute may use to determine any individual record from the table, to his/her medical problem.

One of the methods applied in order to satisfy k-anonymity is the generalisation of data so that the tuples in PT can be distinctly matched to at least k other tuples. Because of the nature of clinical data, we are mainly concerned with hierarchical generalization.

2.1.3 Sensitive-attributes

Sensitive Attributes contain the sensitive person-specific information which an individual will never want to disclose it. Non-Sensitive attributes are those who do not come under remaining three types of attributes.

2.1.4 *k*-anonymity

In the generalized table, a tuple must be indistinguishable from (k-1) other tuples having the same quasi-identifier. A relation is consist of quasi-identifier and non-quasi-identifier attributes in which quasi-identifier attributes needs to be anonymized.

k- Anonymity states that there should be at least k tuples having the same quasi-identifier values to guarantee an individual's privacy [11]. Every tuple in a table should be similar to at least (k-1) tuples then only the table will achieve k-anonymity. k-anonymity is achieved by using generalization and suppression. Following is an example of a table satisfying *2*-anonymity with respect to each attribute [4].

Table 2.2: Example of a data table

Age.	**Gender**	**Zipcode**
34	Male	81667
15	Female	81675
66	Male	81925
70	Female	81931
34	Female	81931
70	Male	81931
45	Male	81931

Table 2.3: Example 2-anonymous data table

Age.	**Gender**	**Zipcode**
< 50	*	816**
< 50	*	816**
≥ 50	*	819**
≥ 50	*	819**
< 50	*	819**
≥ 50	*	819**
< 50	*	819**

2.2 Anonymization

Protection of individual's confidential data is of prime importance. Releasing individual's data (containing sensitive information) publicly might cause risk for individual's privacy [12]. So the first step to anonymize the table is to remove the explicit identifier because this attribute directly reveals identity of record holder.

But L Sweeney's survey [13] shows that removing explicit identifier is not enough to protect individual's privacy. The survey shows that approximately 87 percentage of USA citizens can be re-identified with the help of birth data, zip code and gender attributes when linked with the voter list database to the published medical database. According to this survey, the record holder is linked with the publicly available databases and re-identified with the help of quasi-identifiers (date of birth, gender and age), for this linking attack [9], adversary requires only these two prior knowledge: the record of the victim should be present in the published database and the quasi-identifier of the victim.

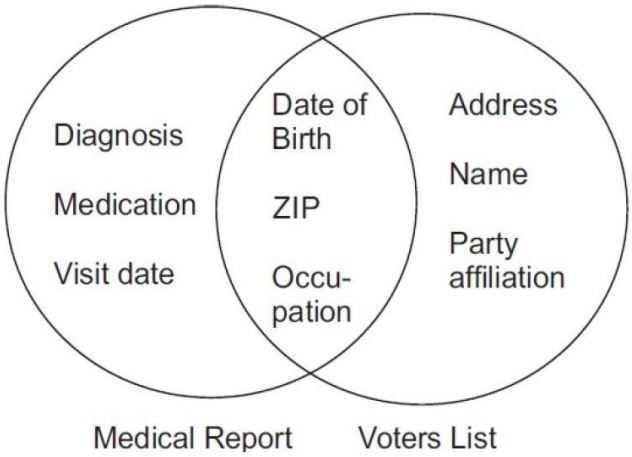

Figure 2.1: Linking attack to identify record holder

2.3 Attack Models in Privacy Preserving and Data Publishing

According to Dalenius [1977] [10], the privacy protection is not allowing an adversary to gain any person-specific sensitive information of a targeted individual even though he has some background knowledge from external sources. The attack models in the PPDP can be categorized in two ways based on their attack principles: [14] In the first type, if an adversary finds a way to map a record holder to a tuple present in the published anonymized table or to an sensitive attribute in the table, these are known as linking attacks. In second type, main focus of the adversary is to gain information about the victim with the help of previously known knowledge (background knowledge).

2.3.1 Record Linkage

Record linkage refers to the mapping of some records to the targeted victim in the publicly released table based on quasi-identifier of the victim. If the victim's quasi-identifier matches with the records in the released table then the adversary faces less no. of possibilities for targeted record with some additional information.

From given tables 2.4 to 2.7, the research centre maps the records in table 2.4 and 2.5 based on same quasi-identifiers present in both table it gain sensitive

information, here by joining these two tables 2.4 and 2.5 for quasi-identifier job, sex and age it can found that male whose age is 38 and profession is lawyer suffers from HIV is mapped to Doug.

To avoid such type of attack by record linkage, a new technique is proposed by Sweeney, Samarati [14] in this model for each set of all quasi-identifiers having same value in table must have at least k number of records .The benefit of this model is that there are other (k-1) tuples that are mapped to same quasi-identifier set with probability of attack $1/k$. As it shown in table 1.1 for quasi-identifier (job, birth, post code).

Subset Property of k-anonymity: If a table is k-anonymous with a set of quasi-identifiers Q, then the must satisfy k-anonymity with respect to all subset Q [15].

Table 2.4: Patient Table

Job.	**Sex**	**Age**	**Disease**
Engineer	Male	35	Hepatitis
Engineer	Male	35	Hepatitis
Lawyer	Male	38	HIV
Writer	Female	30	FLU
Writer	Female	30	HIV
Dancer	Female	30	HIV
Dancer	Female	30	HIV

Table 2.5: 3-Anonymous Table

Job.	**Sex**	**Age**	**Disease**
Professional	Male	35-40	Hepatitis
Professional	Male	35-40	Hepatitis
Professional	Male	35-40	HIV
Artist	Female	30-35	HIV
Artist	Female	30-35	HIV
Artist	Female	30-35	HIV
Artist	Female	30-35	HIV

Table 2.6: External Table

Name.	**Job**	**Sex**	**Age**
Alice	Writer	Female	30
Bob	Engineer	Male	35
Cathy	Writer	Female	30
Doug	Lawyer	Male	38
Emily	Dancer	Female	30
Fred	Engineer	Male	38
Gradys	Dancer	Female	30
Henry	Lawyer	Male	30
Irene	Dancer	Female	32

Table 2.7: 4-Anonymous External Table

Name.	**Job**	**Sex**	**Age**
Alice	Artist	Female	[30-35)
Bob	Professional	Male	[35-40)
Cathy	Artist	Female	[30-35)
Doug	Professional	Male	[35-40)
Emily	Artist	Female	[30-35)
Fred	Professional	Male	[35-40)
Gradys	Artist	Female	[30-35)
Henry	Professional	Male	[30-35)
Irene	Artist	Female	[30-35)

2.3.2 Attribute Linkage

In this attack, attacker gain some information about his sensitive attribute from the released table, even though attacker is not able to link the victim with any individual published record [4]. From the table 2.7, attacker can find that all the female having age 30 whose profession is dance suffer from HIV. So Dance, Female, 30 is confidence 100 percent HIV by this information it found that Emily suffers from HIV. l -Diversity. To prevent from attribute linkage attack it is purposed by Machanavjjhala [16] .Its necessary conditions is every equivalence of released table must have at least l different values. The fundamental concept is to avoid attribute linkage as we seen from the last example if there will be different unique sensitive values it prevents attribute linkage. But probabilistic attacks cannot be

avoided by this because u is very common disease compared to HIV. The released table satisfy l-diverse property if for all qid group:

$$\Rightarrow \sum (P(qid, s) log(P(qid, s)) \geq log(l) \tag{2.1}$$

Here S is sensitive attribute, P(*qid, s*) is a part of records whose sensitive value is s for the total records whose equivalence class is group denoted by *qid* [11]. The more uniformly distributed sensitive values in each equivalence class group qid higher will be the entropy of sensitive attribute. So higher value of entropy in the released table, lesser is the chances probabilistic attack, higher value of threshold l increases its privacy and lesser is the information gain by attacker from released table.

Limitations: The major drawback of entropy l-diversity is it is not able to the measure of probabilistic attack [17] for eg as it is calculated entropy is 1.8 but in second equivalence group out of 4 records 3 suffers from HIV from table 2.7, which is easy for probabilistic attack.

2.4 Anonymizing Operations

The table which contains the original records values of each individual person do not provide any privacy. To publish it and to preserve the privacy of each individual person, some operations have to be performed.

Anonymization is a technique to solve the problem of data publishing, it while keep the sensitive information of record owner which is to be used for data analysis it hides the explicit identity of that record owner from the table which is going to be published.

Anonymization can be done by using following operations [18]

1. Generalization
2. Suppression

2.4.1 Generalization

Generalization modifies the quasi-identifier original most specific value to the some generalized values of specific description [13], e.g. specific form date of birth to generalize to year only while hiding month and date value. Full-domain generalization scheme while generalizing, for all records and for any quasi-identifier, generalization is applied up to few level of hierarchy tree for e.g. If an equivalence class of writer, dancer is generalized to Artist then other equivalence of Engineer, Lawyer must be generalized to Professional. Generalized table is consistent and it is used in global recoding algorithms, but the major drawback of this is data loss is very high..

1. Subtree Generalization

   In subtree generalization scheme [6], at any node other than leaf node, either all its child values are generalized or none is generalized. For example if all dancer is generalized to artist then writer have to be generalized to artist but doctor and engineer may be generalized can retain its specific value at leaf level. It is used in Global recoding algorithms.

2. Sibling Generalization

   In this generalization scheme [7], that is same as subtree generalization but in this some sibling can remain un-generalized. For e.g. If dancer is generalized to artist then writer may remain un-generalized. It gives the lesser distortion compared to subtree and full domain and used in global recoding algorithms.

3. Cell Generalization

   All the generalization schemes [8], that are discussed earlier used, are called global recoding. They give more distortion in this scheme is a value is generalized in one record then for that specific value must be generalized in all other records also. But In cell generalization, it is known as local recoding

there is not restriction means if a value is generalized in one record the same value for same attribute in other record may be un-generalized. For example in a record dancer is generalized to artist dancer in other records may remain un-generalized.

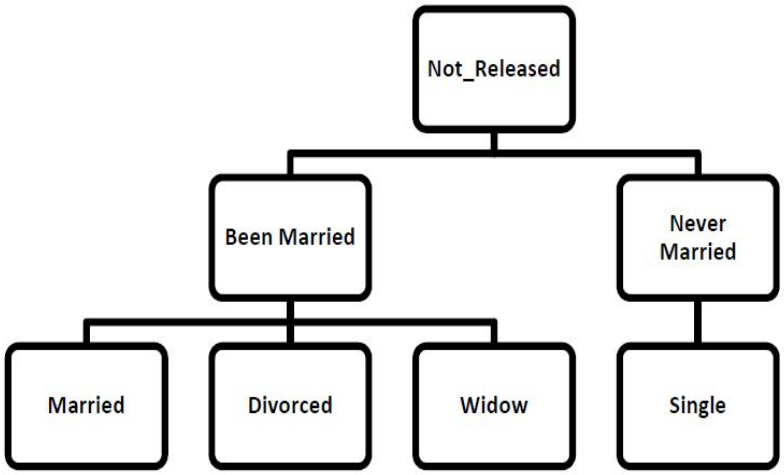

Figure 2.2: GH for marital status

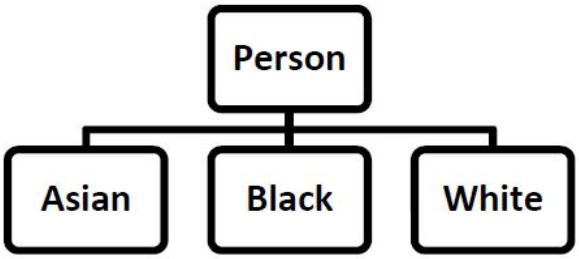

Figure 2.3: GH for Race

2.4.2 Suppression

Suppression is similar to generalization but in this values of quasi-identifier is completely hidden [19] for e.g. from sex male female to any or not released or from specific profession to value is suppressed to not released at all. Different suppression types are defined as

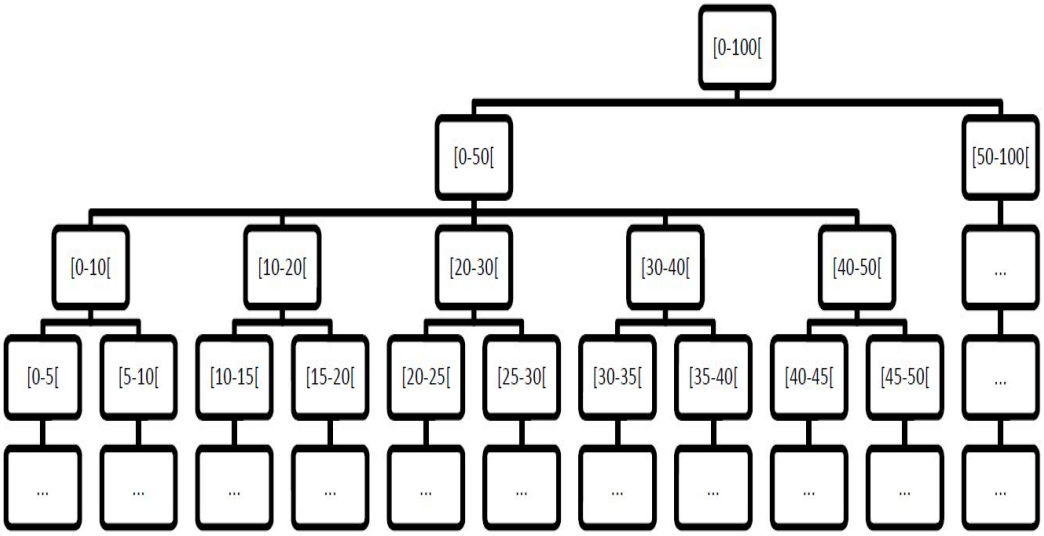

Figure 2.4: GH for Age

1. Record Level: When the complete entry of a record from the table is eliminated or suppressed.

2. Value Level: When all instance or records of a particular value in the table is suppressed

3. Cell Level: When some of records for a given value are suppressed in a table.

2.4.3 Domain Generalization Hierarchy:

Domain Generalization Hierarchy can be defined as a graph or a lattice which acts as the solution space for our k-anonymity problem. The nodes of this lattice are achieved by generalizing different combination of attributes together at various levels. [20]

Example: Consider two attributes "Sex" and "PIN Code" of a relation T. Value of attribute Sex at level 0 of generalization can be "Male" and "Female". To achieve *level-1* of generalization with respect to attribute Sex we must generalize the values "Male" and "Female". We can generalize these two values to another value, say, "Person". By generalizing the values of attribute Sex to "Person" we achieve *level-2* generalization with respect to Sex. Lets take another attribute PIN

Code from relation T. Let us assume that PIN Code can have values "110010", "110011" and "110012" at *level-0* generalization. We can generalize these values to "11000x" and "11001x" to achieve level 1 generalization with respect to attribute PIN Code. Further, we can generalize the values to "1100xx" in order to achieve *level-2* generalization with respect to attribute PIN Code. By combining different levels of generalization of different attributes we can form the Domain Generalization Hierarchy as shown in the fig 2.5.

Full Domain Generalization Hierarchy: can be defined as a graph or a lattice which acts as the solution space for our k-anonymity problem. The nodes of this lattice are achieved by generalizing different combination of attributes together at various levels.

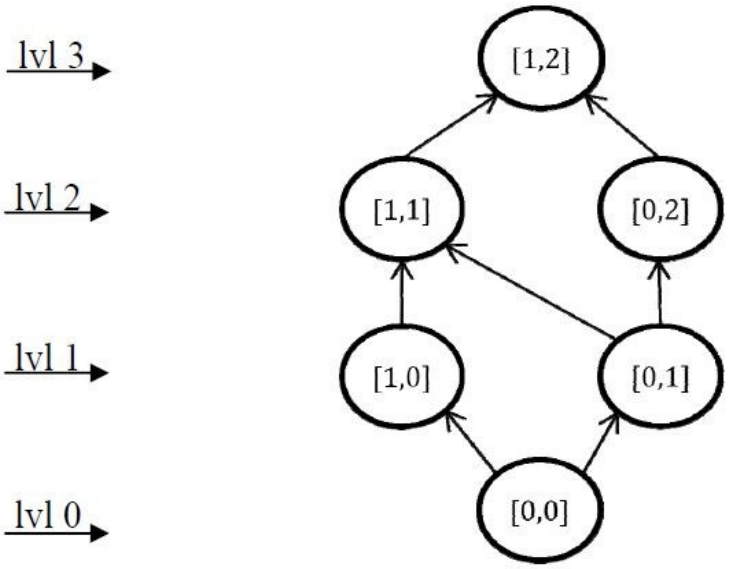

Figure 2.5: Example of a Lattice generalization

2.5 Metrics used to Measure the Quality of Generalized Data

Privacy preserving data publishing have two objectives, privacy of individual entity for each record must be preserved and published data must be information which is useful for data mining. So the quality of anonymized data can be measured by data metric which are classified into three categories.

2.5.1 General Purpose Metrics:

When data publisher do not know what data recipient want to know or analysis from the published data so data publisher cant focus on any particular data utility [15]. In this case data published is open to all like internet so that data recipient based on their different interest and they do data mining according to their requirement, in this is very obvious that same metric is not good or accurate for different recipients. In this case for better utility of anonymized data, data publisher choose metric which are more suitable for mostly all data recipients such as ILoss, distortion, discernibility.

2.5.2 Special Purpose Metrics

If data publisher know for which purpose the published data will be data mined or in which information or pattern data recipient is interested, so that they can preserve their related information and publish the data according to their requirements. For example if the purpose of data recipient is to model the classification based on a particular attribute in this case generalization must not be done for values whose identification is necessary to assign a class, which is used for their classification [21].

Classification Metric (CM) Iyengar proposed a metric to measure the classification error means a record is assigned to a class by assuming that in it a particular class is not majority but in reality that class is not the majority class so, record is assigned to wrong class [11]. There must be some penalty for it or there is a penalty if record is suppressed completely and not assigned to the

any class. CM can be calculated by sum of all the penalties of each record, it is normalized by considering total number to records.

$$CM = \frac{\sum_{allrows} Penalty(row_r)}{N} \tag{2.2}$$

A row_r is given penalty if the row is suppressed and/or if its class label class(r) is not the majority class label majority (G) of its group G.

Penalty can be calculated as if a record is suppressed or it is assigned to group assume class(r) is major class but actual that class is not the major class.

2.5.3 Trade-off Metrics

Specializing from a general value to a specific value loss some level of privacy but gain some information regarding that attribute which is specialized. Special metric while anonymizing at final information it may gain sufficient information but might lose so privacy that it is very difficult to do further anonymization. So Trade-off Metrics solve this problem, both information gain and privacy loss are calculated at every iteration of anonymization, so that optimal trade -off can be found for both necessary requirements.

In this trade-off metric, for every specialization all records of this group are assigned to its child level group so it gain some information(IG) and as it divides the group size into smaller group there is privacy loss(PL). Objective of this metric is to find a specialization whose information gain is maximum for each privacy loss

$$IGPL = \frac{IG(s)}{PL(s) + 1} \tag{2.3}$$

Where IG(s) = Information gain can be decrement of class entropy or decrement of distortion by specialization.

$$PL(s) = avgA(QID_j) - A_s(QID_j) \tag{2.4}$$

Where Privacy loss PL(s) is the average decrement of anonymity over all QID_j that contain the attribute of s and

A(QID_j) = the anonymity before specializing of attribute j.

2.6 Global Recording Algorithms

2.6.1 Datafly Algorithm

The Datafly algorithm [Sweeney (1997)] goes with the assumptions that the best solutions are the ones that are attained after generalizing the variables with the most distinct values (unique items) [1]. The search space is the whole lattice. However, this approach only goes through a few nodes in the lattice to find its solution. This approach is very efficient from a time perspective. Datafly uses a greedy algorithm to search the domain generalization hierarchy. At every step, it chooses the locally optimal move. One drawback with Datafly approach is that it may become trapped in a local optimum.

Here is a summary of the Datafly algorithm:

1. Consider a table MT = PT[QI] (takes into consideration only the quasi-identifiers fields)
2. While k-anonymity is not achieved and the count of the remaining rows that do not comply to k-anonymity is more than k:
   (a) Get the number of distinct values of each attribute in MT
   (b) Generalize the attribute with the most distinct values
3. Suppress the remaining rows

2.6.2 Samarati Algorithm

Samarati algorithm assumes that the best solutions in the lattice are the ones that result in a table having minimal generalizations [10]. So, the solutions are available in the height that is minimal in a lattice. The algorithm is based on the axiom that if a node at level h, in domain generalization hierarchy satisfies k-anonymity, then all the levels of height higher than h also satisfy k-anonymity. In order to search the lattice and identify the lowest level with the generalizations that satisfy k-anonymity with minimal suppression, Samarati used binary search. The

algorithm goes through the lattice with a binary search, always cutting the search space in half. It goes down the level if a solution is found at that level, otherwise it goes up the lattice [12]. Eventually, the algorithm finds the solution with the lowest height with the least generalizations. This level ensures less information loss but time consumed is higher than Datafly.

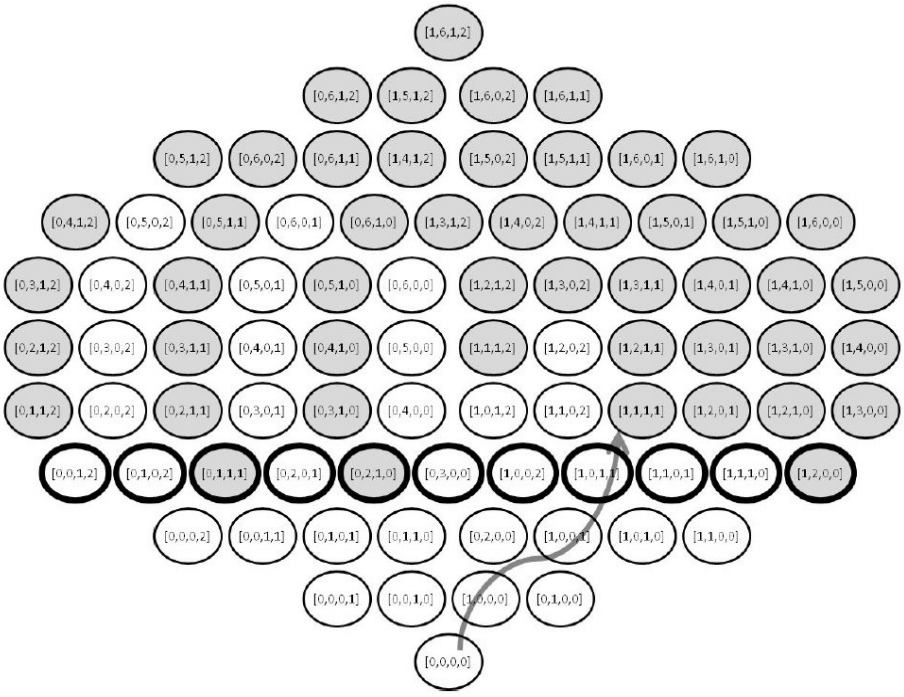

Figure 2.6: Visual comparison of Datafly and Samaratis algorithms

The summary of Samarati algorithm:

1. Consider a table T = PT[QI] to be generalized (takes into consideration only the quasi-identifiers fields).

2. Consider the middle height in the area of search (area of search is initially the whole lattice).

3. Check if at that height there is at least one node that satisfies k-anonymity with minimum suppression (the minimum suppression variable would be already set) then,

(a) If not the minimum, specify the upper half as the new area of search.

(b) If minimum, specify the lower half as the new area of search.

4. If the area of search consists of more than one level in the lattice, repeat step 2. Otherwise, return a solution residing on this level.

2.6.3 Incognito Algorithm

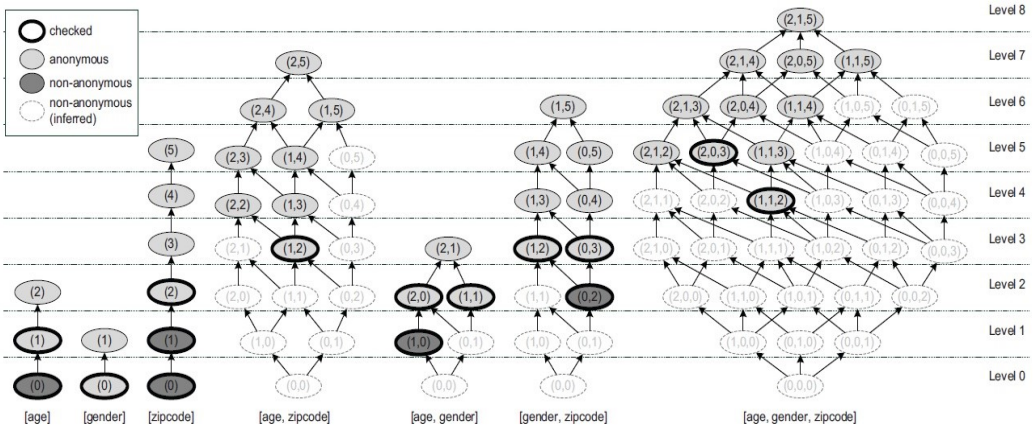

Figure 2.7: Example of Incognito algorithm

Incognito implements a dynamic programming approach which satisfy subset property which states that a relation T cannot be k-anonymous if it's subset of quasi-identifiers does not satisfy k-anonymity. The approach constructs generalization lattice of each subset of QIs and checks by performing a breadth-first bottom-up search [18]. The number of generalization lattice constructed in case of Incognito for QIs of order r is 2r. Thus Incognito algorithm is of order (2r) because at least one lattice is checked for k-anonymity in every generalization lattice.

2.6.4 OLA Algorithm

El Emam et al: suggested an algorithm called Optimal Lattice Anonymization and presented that it outperforms Incognito [20]. It use predictive-tagging to reduce the search space of the lattice. However, if global optimal k-anonymous lattice lie on or above the middle level of full domain generalized hierarchy, then

the algorithm check all the middle level lattices for k-anonymity. This algorithms checks only the middle level of full domain generalized hierarchy is exponential in number of QIs.

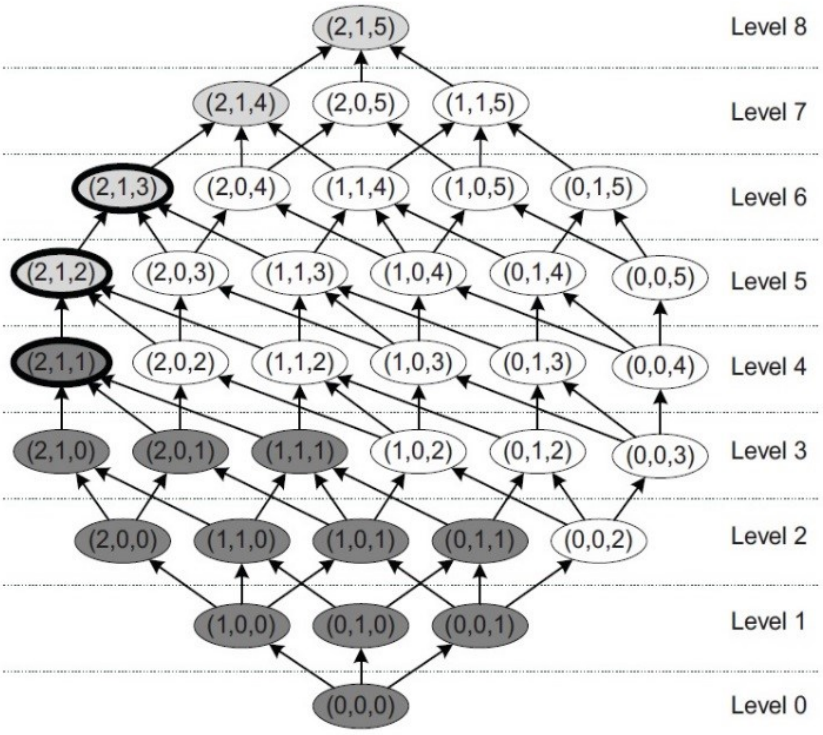

Figure 2.8: Example of OLA algorithm

Chapter 3

Proposed Work

3.1 Introduction

k- Anonymization is a primary approach for the de-identification of datasets containing person specific information. In our work, we have described a approach to implement most of the k-anonymity algorithms and also proposed a parallel scheme that produces better results with real-world datasets. The maximum count of QIs for the datasets considered by them is only nine. If the count of QI's is very high, then it would be difficult to put all the data items in the main memory.

1. **ILoss** To calculate the data loss while anonymizing the data proposed a data metric known as ILoss [12].

$$ILoss = \frac{||v_g|| - 1}{||D_A||} \tag{3.1}$$

Where $||v_g||$ is total number of childrens of the node.

$||v_g||$ is total count of leaf nodes for that attribute having Vg as a node. If ILoss= 0, means value remains ungeneralized, same as in original table. It calculates the fraction of leaf nodes that are generalized.

Example: Let a value is generalized from Lawyer to professional. So its ILoss $=\frac{2-1}{4} = 0.25$. After generalization ILoss for any record can calculated as

$$ILoss(r) = \sum(W_g * ILoss(v_g)) \tag{3.2}$$

Where W_g is predefined weight penalty assigned to each quasi-identifier The total for complete generalized table is

$$ILoss(r) = \sum_{r \in T} ILoss(r) \tag{3.3}$$

2. **Discernibility** After anonymizing dataset, each equivalence class has its size that is number of records in it. The class size contributes to the anonymization based on cost, it can be calculated for complete generalized dataset by using the formula

$$DM = ||E_i||^2 \tag{3.4}$$

Where $||E_i||$ is the size of equivalence class minimize Discernibility cost leads to less distortion with is desirable requirement for better anonymization.

3.2 Proposed Approach

Our work is based on a general framework for the efficient application of k-anonymity based algorithms. In [21], suggested a time efficient application of the k-anonymization algorithm. Furthermore, we evaluate the framework in current section and outline the fundamental objective behind it. The main task is to check the k-anonymous status of level nodes in a parallel manner by using the threads and this task should be time efficient.

The preliminary work of this scheme is a well-planned memory layout, which allows the optimal application of various generalization schemes to a given dataset. Additionally, the anonymization operations are problem specific. It offers some further optimization. The general implementation, involving optimization applied to all global recording based anonymization schemes i.e. samarati algorithm.

Algorithm 1 SearchLattice

Input: *GeneralizedLattice, minLevel, maxLevel*
Output: *optimalNode*

```
1: if minLevel > maxLevel then
2:    optimalNode ← minILoss(anonymityNodes);
3:    return optimalNode;
4: else
5:    midLevel ← ⌊(minLevel+maxLevel)/2⌋;
6:    anonymityNodes ← SearchNodesParallel(GeneralizedLattice, midLevel);
7:    if |anonymityNodes| > 0 then
8:       SearchLattice(GeneralizedLattice, minLevel, midLevel-1);
9:    else
10:      SearchLattice(GeneralizedLattice, midLevel+1, maxLevel);
11:   end if
12: end if
```

The generalized lattice with minimum level and maximum level i.e. height will be given to the above function which acts as a main program for finding the optimal node, we use the binary search method. The *midLevel* will be calculated by taking the half of the lattice height and then the Lattice will be passed through the *SearchNodesParallel()* to perform the parallelism for the nodes in that level.

The *SearchLattice( )* uses the recursive method by calling itself. Thus if there is at least one node in *anonymityNodes(midLevel)* then the recursion will take place as *SearchLattice(minLevel, midLevel-1)* i.e. we will consider the lower part of the lattice with respect to *midLevel*. Otherwise the *anonymityNodes(midLevel+1, maxLevel)* i.e. we will consider the upper part of the lattice with respect to the *midLevel*. When the *minLevel* is greater than the *maxLevel* we finally calculate the information loss for all the nodes in the *minLevel* that is having at least having one node in the lattice and it will assign to the *optimalNode* and it will return as output.

Algorithm 2 SearchNodesParallel

Input: *GenerlizedLattice, midLevel*
Output: *anonymityNodes*
1: *N* ← no. of nodes at *midlevel*;
2: *Node[N]* ← store nodes from *midLevel*;
3: **parfor** *i = 1 to N*
4: **if** *kAnonymous(Node[i])* == TRUE **then**
5: *anonymityNodes[ ]* ← *Node[i]*;
6: **end if**
7: **end parfor**
8: **return** *anonymityNodes*;

We use parallelism to evaluate nodes of the level through parallelism. The *SearchNodesParallel( )* will take the parameters level which will have the nodes in it and the lattice. Create a thread for the each node in that level, each thread will execute parallel in the run. The nodes which will satisfy k-anonymity will assign to the *anonymityNodes( )*. The output will be generated the *anonymityNodes( )*.

3.3 Basic Implementation

Now we apply our algorithm to the three quasi-identifier of Age, Gender, Zipcode as shown in the table 2.2. The maximum generalization gierarchied for the Age=2, Gender=1 and Zipcode=5.

Now for k=2, the algorithm follows the below steps to find the level to apply parallelism, so all the nodes in that level can execute parallel. The *Generslized-Lattice* is given in the fig 3.1 acts as a input search space.

The solution space of lattice as shown in the figure 3.5, is having the *minLevel* =4 and the *optimalNode* is (1,1,2)

The solution space of lattice as shown in the figure 3.9, is having the *minLevel* =5 and the *optimalNode* having the nodes are (2,0,3) and (1,1,3). So now we need to calculate the $ILoss$ for the both nodes as given in the equation 3.2 and return the *optimalNode* as the node which is having minimum *ILoss*.

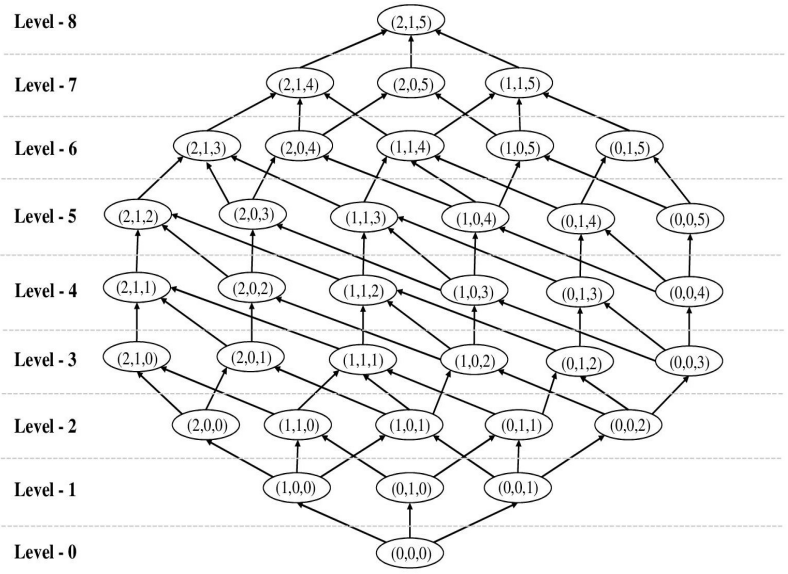

Figure 3.1: Generalized Lattice

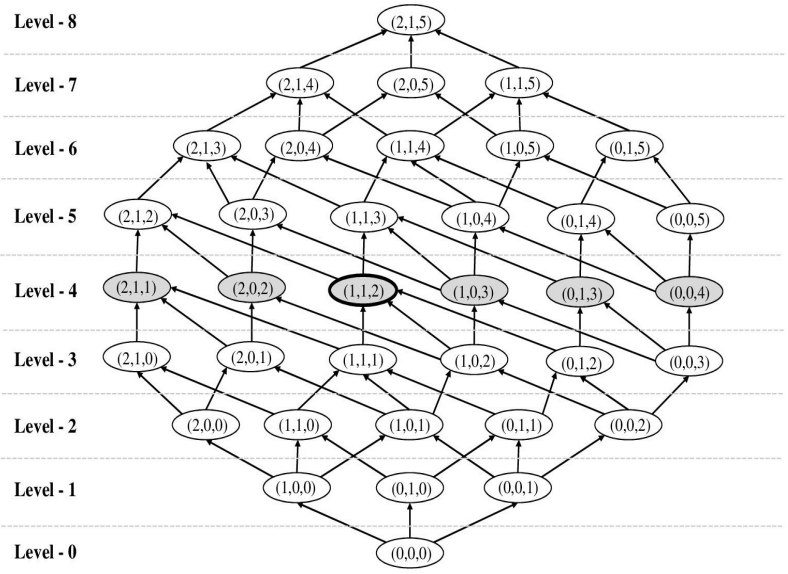

Figure 3.2: Step-1 Lattice with k=2

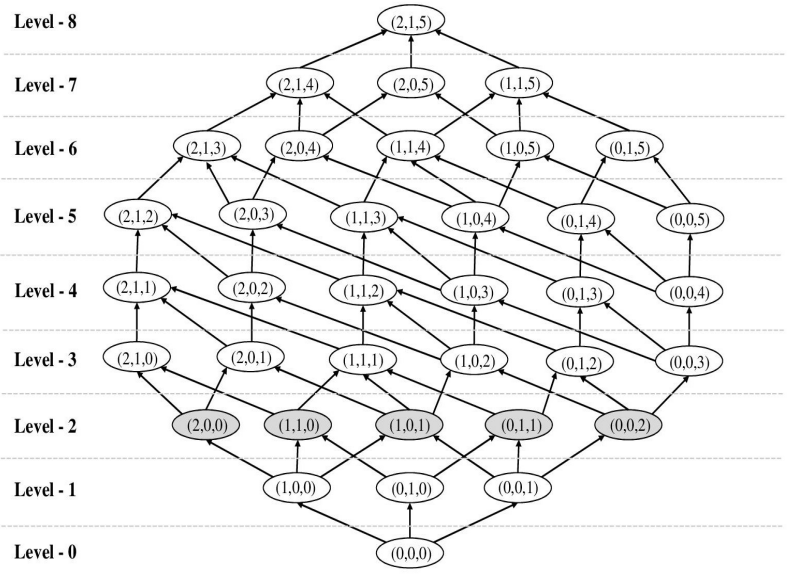

Figure 3.3: Step-2 Lattice with k=2

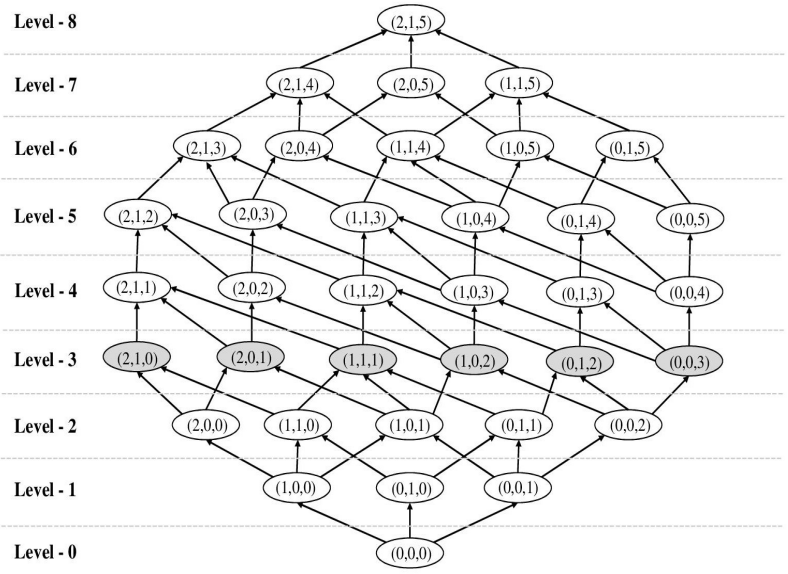

Figure 3.4: Step-3 Lattice with k=2

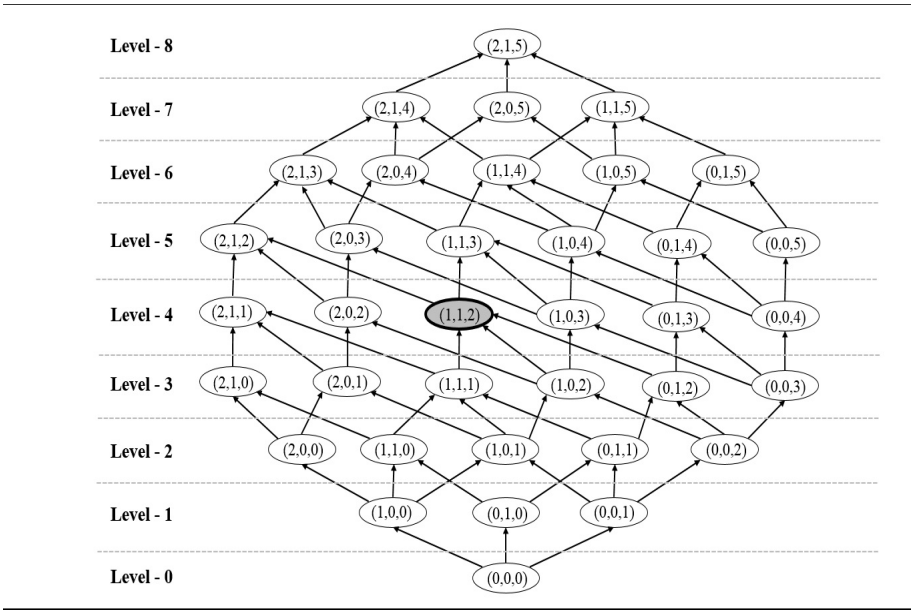

Figure 3.5: Solution space of Lattice with k=2

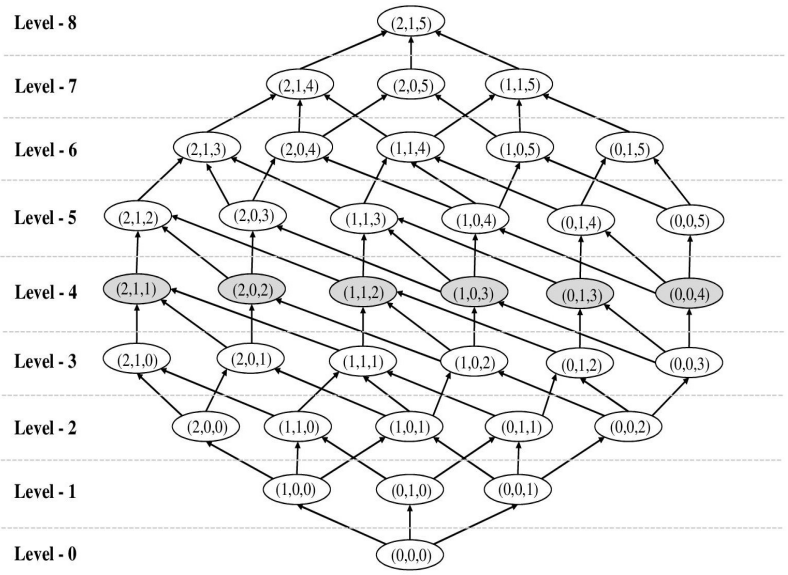

Figure 3.6: Step-1 Lattice with k=5

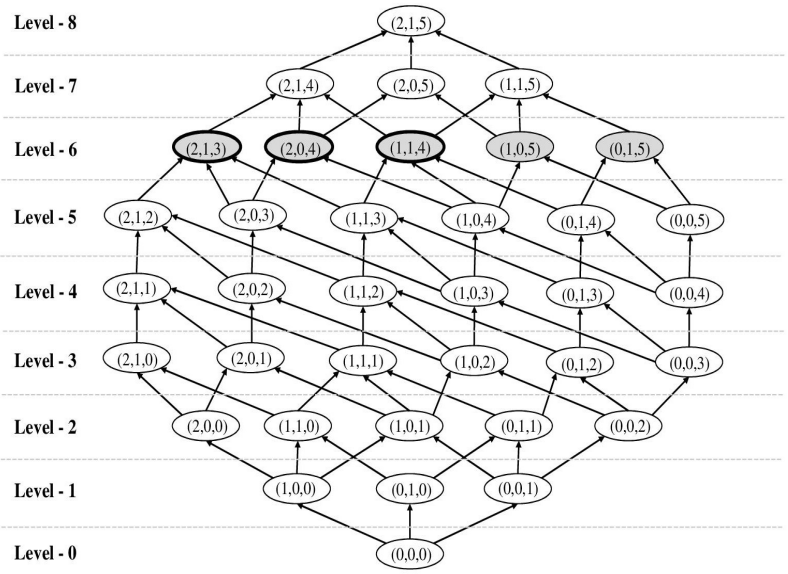

Figure 3.7: Step-2 Lattice with k=5

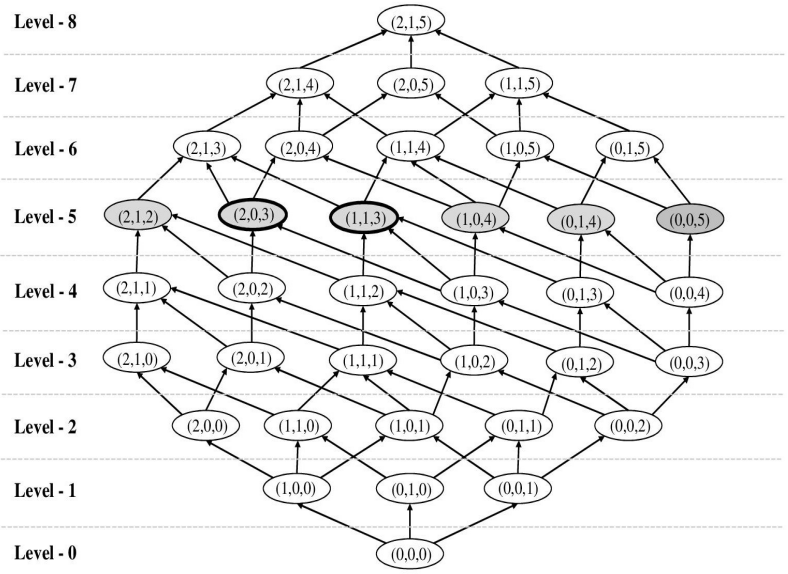

Figure 3.8: Step-3 Lattice with k=5

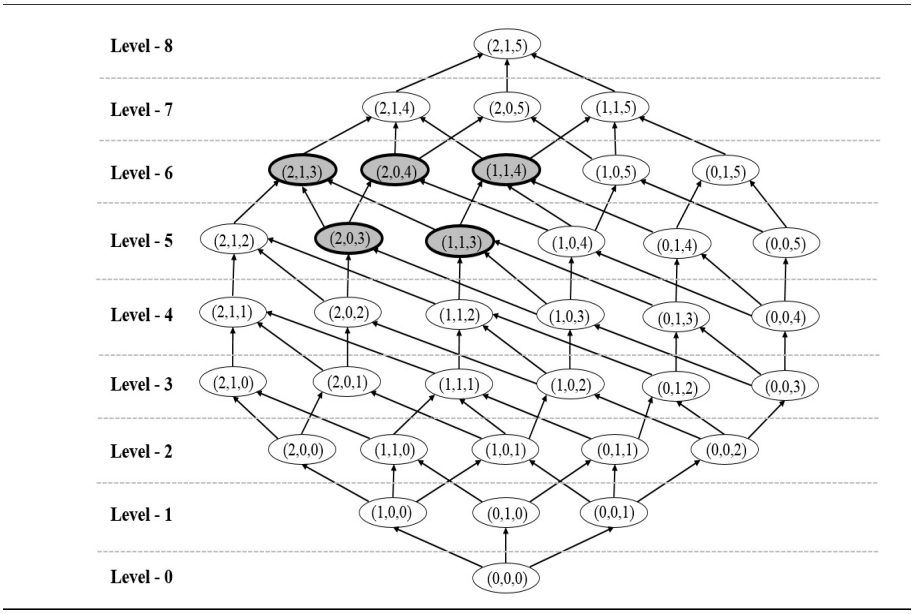

Figure 3.9: Solution space of Lattice with k=5

Chapter 4

Implementation and Results

4.1 Implementation

The data was extracted from the CENSUS data set which is available at `ftp://ftp.ics.uci.edu/pub/machine-learning-databases/adult/`. The data consists of 7 attributes (age, gender, education level, marital status, race, work class, country). The number of data records are 31,062.

Implementation is done on System having configuration Intel (R) core(TM) i7-2670QM CPU @ 2.20 GHz, 8GB RAM. Our implementation is done JAVA. Complete Adult Data Set which contains 31,062 records is taken for analysis results. The attributes for quasi identifier are Age is numeric, Work class is categorical, Education is categorical, Marital status is categorical, Race is categorical, Gender is categorical, and Occupation and Salary are sensitive attributes. We have taken Discernibility Metric and Execution Time as parameters to evaluate and analyses the result for k values taken as 2, 5, 10 over the proposed algorithm and other previous algorithms like Samarati, Incognito, OLA(Optimal lattice Anonymization).

Table 4.1: Description of Adult Dataset

S.No.	Attributes	Generalizations	Distict value	Height
1	Work Class	Taxonomy Tree	7	3
2	Education	Taxonomy Tree	16	4
3	Marital Status	Taxonomy Tree	7	3
4	Race	Taxonomy Tree	5	2
5	Sex	Suppression	2	1
6	Occupation	Taxonomy Tree	14	2
7	Salary	Suppression	2	1

4.2 Results

4.2.1 Discernibility Metric

We used Discernibility Metric to measure the quality of anonymized data, the lesser is discernibility cost, better is the quality is anonymized Data. By referring figures Figure-4.1, Figure-4.3 and Figure-4.5, we can conclude that For smaller k value k=2, 5 and 10, for all number quasi-identifiers taken our approach give better anonymized data than incognito, Samarati and OLA algorithm and if k is large, k= 10 and number of quasi identifier taken not large our approach gives lesser discernibility.

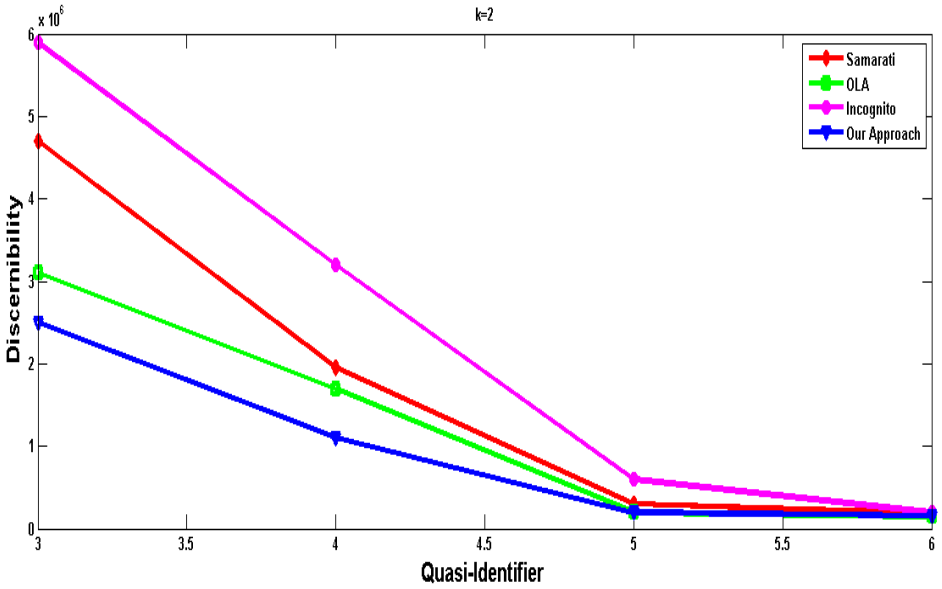

Figure 4.1: Discernibility vs Quasi-Identifier

4.2.2 Execution Time

We considered Execution time also to evaluate and compare our approach with Incognito and Samarati and OLA. By referring figures Figure-4.2, Figure-4.4, Figure-4.6, we can conclude that for all k values 2, 5, 10 and our approach take lesser execution time than Incognito, Samarati, and OLA algorithm. For all k values taken and for all number of quasi identifier taken so we can conclude our approach is faster compared to others.

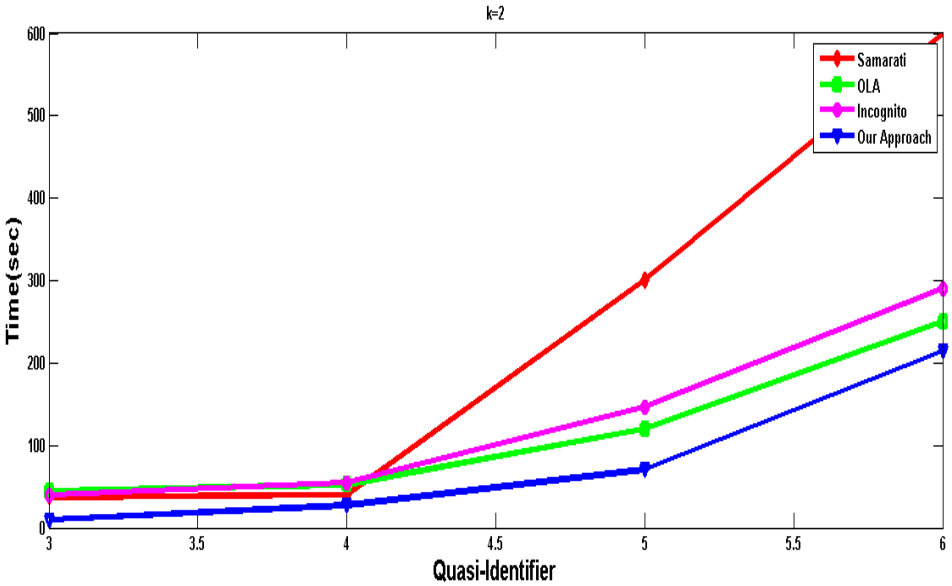

Figure 4.2: Time(sec) vs Quasi-Identifier

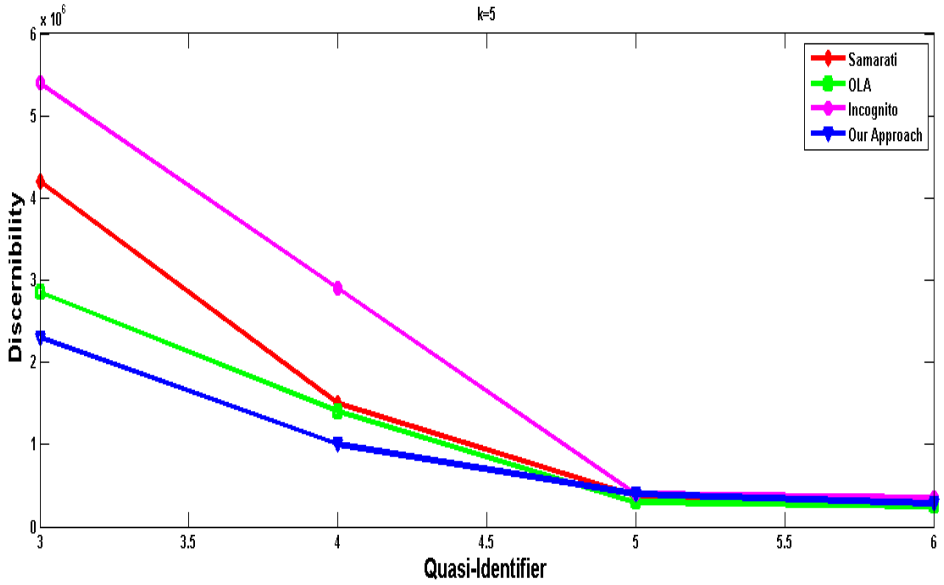

Figure 4.3: Discernibility vs Quasi-Identifier

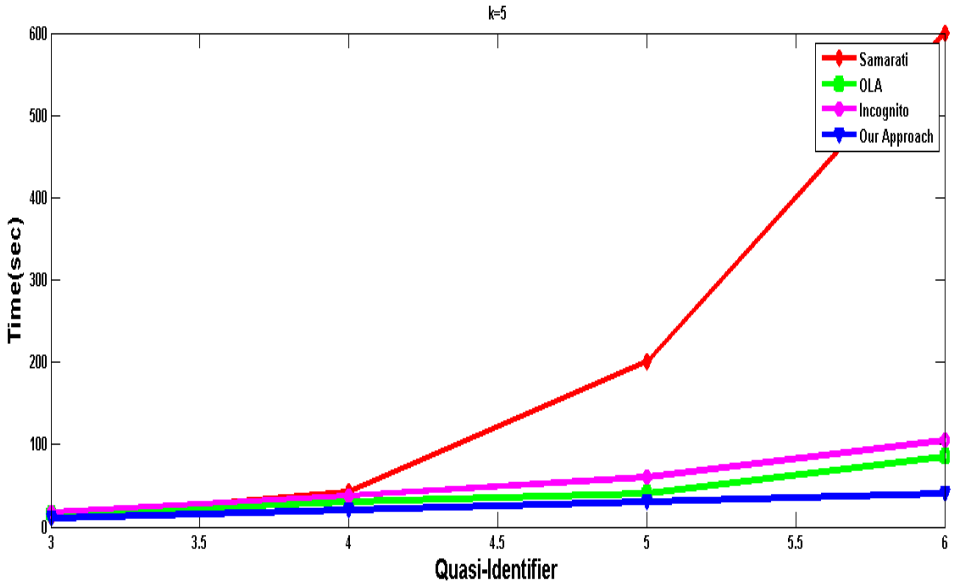

Figure 4.4: Time(sec) vs Quasi-Identifier

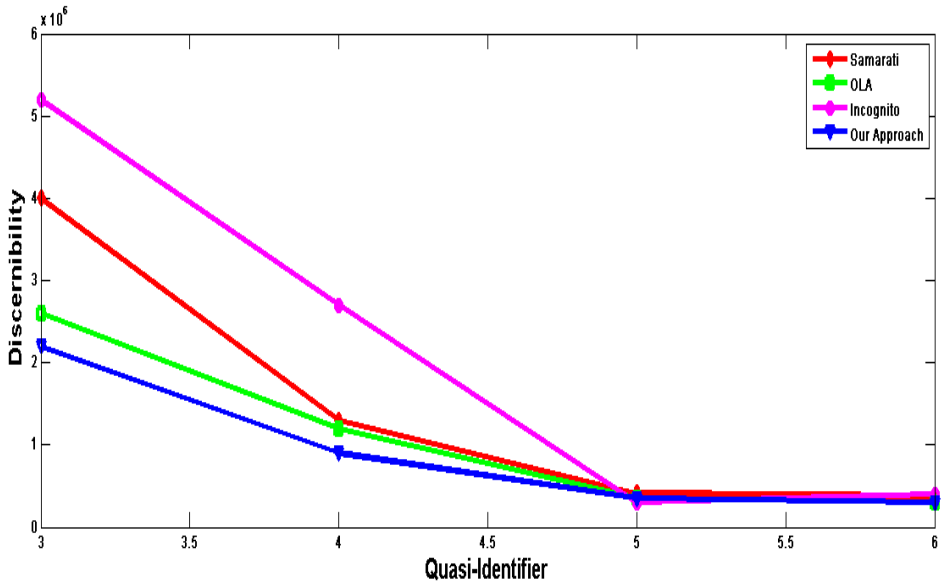

Figure 4.5: Discernibility vs Quasi-Identifier

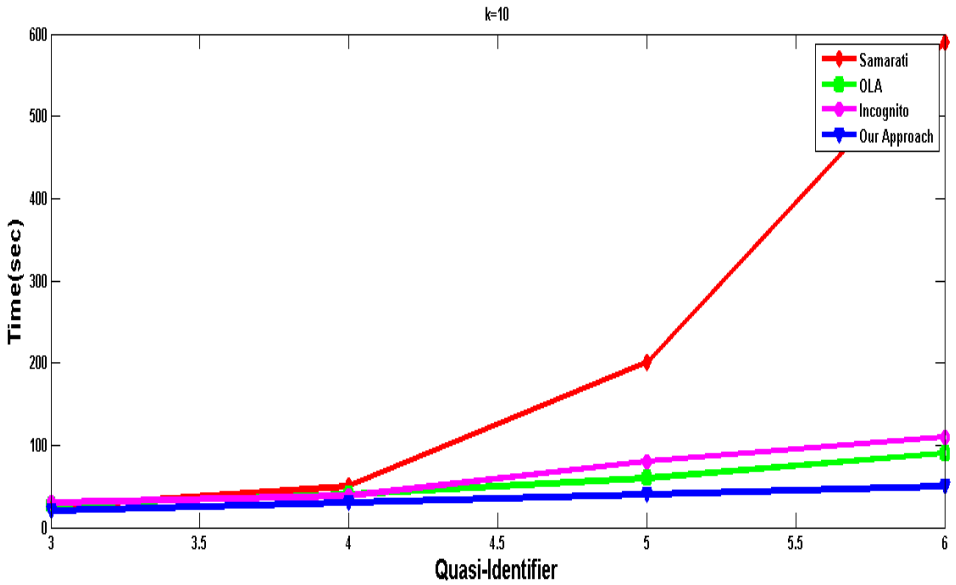

Figure 4.6: Time(sec) vs Quasi-Identifier

Chapter 5

Conclusion and Future Work

In our work, we have described a framework to implement most of the k-anonymity algorithms and also proposed a parallelism scheme that produces better results with real-world datasets. We explained that the frameworks applicable for the implementation of k-anonymity algorithms like Incognito, Datafly and optimal lattice anonymization (OLA). The proposed approach of k- anonymization, which gives better result than Incognito, Samarati, OLA, and Datafly. As it traverse the lattice through binary search method, and it utilizes the parallelism in best way to reduce the time complexity extensively with the use of the proposed layout.

In future, the algorithms discussed in this thesis can be further improved by reducing the size of the solution space and applying improved searching algorithms.

Bibliography

[1] P. Samarati and L. Sweeney, "Generalizing data to provide anonymity when disclosing information," in *PODS*, vol. 98, p. 188, 1998.

[2] B. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Computing Surveys (CSUR)*, vol. 42, no. 4, p. 14, 2010.

[3] Y. Yuan, J. Yang, J. Zhang, S. Lan, and J. Zhang, "Evolution of privacy-preserving data publishing," in *Anti-Counterfeiting, Security and Identification (ASID), 2011 IEEE International Conference on*, pp. 34–37, IEEE, 2011.

[4] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[5] P. Shi, L. Xiong, and B. Fung, "Anonymizing data with quasi-sensitive attribute values," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 1389–1392, ACM, 2010.

[6] S. K. Adusumalli and V. V. Kumari, "Attribute based anonymity for preserving privacy," in *Advances in Computing and Communications*, pp. 572–579, Springer, 2011.

[7] B. Berčič and C. George, "Identifying personal data using relational database design principles," *International Journal of Law and Information Technology*, vol. 17, no. 3, pp. 233–251, 2009.

[8] K. El Emam, F. K. Dankar, R. Issa, E. Jonker, D. Amyot, E. Cogo, J.-P. Corriveau, M. Walker, S. Chowdhury, R. Vaillancourt, *et al.*, "A globally optimal k-anonymity method for the de-identification of health data," *Journal of the American Medical Informatics Association*, vol. 16, no. 5, pp. 670–682, 2009.

[9] J. Goldberger and T. Tassa, "Efficient anonymizations with enhanced utility," in *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, pp. 106–113, IEEE, 2009.

[10] M. Hua and J. Pei, "A survey of utility-based privacy-preserving data transformation methods," in *Privacy-Preserving Data Mining*, pp. 207–237, Springer, 2008.

[11] G. V. Kanth and B. S. Kumar, "A study of novel anonymization techniques for secure data publishing," in *International Journal of Engineering Research and Technology*, vol. 2, ESRSA Publications, 2013.

[12] A. Gionis and T. Tassa, "k-anonymization with minimal loss of information," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 2, pp. 206–219, 2009.

[13] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 571–588, 2002.

[14] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," tech. rep., Technical report, SRI International, 1998.

[15] B. C. Fung, K. Wang, A. W.-C. Fu, and S. Y. Philip, *Introduction to privacy-preserving data publishing: concepts and techniques*. CRC Press, 2010.

[16] V. S. Iyengar, "Transforming data to satisfy privacy constraints," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 279–288, ACM, 2002.

[17] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient full-domain k-anonymity," in *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pp. 49–60, ACM, 2005.

[18] A. Meyerson and R. Williams, "On the complexity of optimal k-anonymity," in *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 223–228, ACM, 2004.

[19] J. Gehrke, "Models and methods for privacy-preserving data analysis and publishing," in *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pp. 105–105, IEEE, 2006.

[20] R. C.-W. Wong, A. W.-C. Fu, K. Wang, and J. Pei, "Anonymization-based attacks in privacy-preserving data publishing," *ACM Transactions on Database Systems (TODS)*, vol. 34, no. 2, p. 8, 2009.

[21] Z. FeiFei, D. LiFeng, W. Kun, and L. Yang, "Study on privacy protection algorithm based on k-anonymity," *Physics Procedia*, vol. 33, pp. 483–490, 2012.